PROTOTYPING IN PRACTICE: THE CHEAT SHEET // ERICA HEINZ

Useful prototypes are:
- fast
- non-precious
- interactive

Why prototype?
- work in real medium: interactions
- test your riskiest assumptions

Example: The 5 part landing page
- unique selling points
- hero shot
- benefits
- social proof
- call to action

Research
- know your users:
  - role wants goal and needs user experience
- user interviews
- expert interviews
- group interviews
- self documentation
- in-context immersion
- community-led discovery
- parallel inspiration
- competitive research
- surveys
- problem interviews

Testing
- solution interviews
- concept testing
- usability testing
- A/B testing
- analytics
- AARRR metrics

Process
- sprint: think > make > check
- try a flex week after each sprint, or stagger design & dev

Challenges
- people problems? require respect
- feedback flood? take quiet time
- miss hifi? let go, embrace 80%

WRITING (test conceptual risks)
- clarify concept for self, team, product, public
- test synonyms, must match mental model

WHITEBOARD (test conceptual risks)
- good for quick iterations
- requires initiative, communication
- negligible attachment
- pro: big canvas, egalitarian
- con: non-interactive

PAPER (test experiential, holistic risks)
- solve what's on the page, where
- good for teams, open thinking
- requires co-location, chutzpah
- low attachment
- pro: fun, super fast, approachable
- con: slow cut and paste, no remote access

BITMAP (test strategic, experiential risks)
- see how things feel
- good for investors, clients, testers, designers
- require design skills and programs
- moderate attachment
- pro: crystal clear, familiar
- con: kind of precious, maintenance is a pain

VECTOR (test holistic, experiential risks)
- work out how flows fit together
- good for teams, some clients, text-heavy apps
- require layout skills and programs
- low attachment
- pro: symbol libraries, clickable if needed
- con: left-brained

CODE (test technical, experiential risks)
- demonstrate how things work
- good for team, testers, some clients, coders
- require coding skills
- moderate attachment
- pro: clicks, hovers, gifs!, tested, responsive
- drawbacks: left-brained, preformatted

Which kind?
- what do you need to test? why?
- who's the audience? what'll they understand?
- what are your skills? where should they be?
- how much time/money can you spend?

Written prototyping
- whiteboard*
- sticky notes*
- Wordnik.com

Paper prototyping
- paper, markers*
- Jot
- Balsamiq
- Prototyping On Paper
- Prott

Bitmap/Vector prototyping
- Google Drawing*
- Sketch*
- Invision*
- Axure
- Keynote
- Macaw
- Marvel
- Pixate
- Proto.io
- Prott
- UXPin
- Webflow

Code prototyping
- Bourbon*
- Bootstrap
- Foundation
- Angular JS
- Ember JS
- React JS
- Framer
- Xcode / Quartz Composer
- Origami

*Tools I use frequently

This niche changes rapidly, clear leaders among these tools have not yet emerged. To keep up, follow newsletters/sites like:
- Unmatched Style
- Responsive Design Weekly
- UX Design Weekly
- Design Tools News (on Khoi Vinh's blog)
- TodoMVC.com